UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/713,343	11/14/2003	Claude Basso	RPS920030063US1	9860

45211 7590 03/15/2007

Robert A. Voigt, Jr.
WINSTEAD SECHREST & MINICK PC
PO BOX 50784
DALLAS, TX 75201

EXAMINER	
FEARER, MARK D	
ART UNIT	**PAPER NUMBER**
2109	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/15/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)
	10/713,343	BASSO ET AL.
	Examiner	Art Unit
	Mark D. Fearer	2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 November 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *November 14, 2003*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Information Disclosure Statement

The information disclosure statement submitted on 14November2003 has been considered by the Examiner and made of record in the application file.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-5 and 8-12 are rejected under 35 U.S.C. 102(e) as being anticipated by Garnett et al. (US 7032037 B2).

Consider claim 1. Garnett et al. clearly shows and discloses a method for reducing the number of messages to be processed by a control processor in a load balancer comprising the steps of: receiving a request to establish a TCP connection from a client by a network processor in said load balancer (("The network processing unit 520 further includes packet processing circuitry 522 to classify packets at layer 4 and look up flow details of that packet. If a flow exists, the network processing unit 520 modifies and forwards the packet to continue the flow.") column 39 lines 48-52);

establishing said TCP connection with said client via handshake messages between said network processor and said client ((“For new TCP flows that are to be application layer load balanced, it performs the TCP three-way handshake and forwards the request data packets to an application layer classification engine 528. The network processing unit 520 also handles handover of a newly established connection to a selected server 43.”) column 39 lines 52-58); receiving a request message from said client ((“Transport layer load balancing in a TCP/IP network can be performed on receipt of a TCP synchronize/start packet (TCP SYN) from a client to a specific destination.”) column 35 lines 9-11); bundling said request message and information from said handshake messages involved in establishing said TCP connection by said network processor ((“In HTTP, usually the first data packet contains the request so the first data packet, after the three-way handshake, is parsed to make the load balancing decision. After the load balancing decision has been made, the connection to the load balancer is handed over to the selected server.”) column 36 lines 52-56); and transmitting said bundled message to said control processor by said network processor (“When a server responds to a received data packet, the outgoing packet is transmitted from the server 505 to the Load Balancer 501, which forwards the outgoing packet to the outside network. Shown in FIG. 20b is a representation of the data packet paths through a shelf 41 arranged as shown in FIG. 19b. Here, the incoming packet arrives at the switch 73, travels to the load balancer 501 and is then passed to a processing cartridge 43 via the switch 73.”) column 33 lines 13-20).

Consider claim 2, and as applied to claim 1 above. Garnett et al. discloses a method comprising the steps of: identifying a server in a server farm to service a client's request message by a control processor ((“Management control of the web farm 360 can be provided through a pair of System Management Servers (SMSs) 362.”) column 29 lines 23-24); bundling the client's request message and a control message by the control processor (“Thus incoming data packets arrive at the load balancer and are routed through to a selected server 505”) column 32 lines 63-65); and transmitting the bundled message comprising the client's request message and the control message to a network processor (“A control microprocessor 530 performs management tasks for the load balancer 501.”) column 40 lines 31-32).

Consider claim 3, and as applied to claim 2 above. Garnett et al. discloses a method wherein said server in said server farm is identified using information extracted from said client's request message ((“An information distribution module may be provided removably received in the carrier operable connect to the internal communications network to receive an information message, to perform processing on the message to determine a destination, and to forward the message toward the determined destination via the internal communications network. This arrangement provides a modular computer system with integral load balancing service.”) column 1 lines 53-56).

Consider claim 4, and as applied to claim 2 above. Garnett et al. discloses a method wherein said control message comprises information used to enable said network processor to create entries in a forwarding table to ensure packets from said

client are transmitted to said server and to ensure packets from said server are transmitted to said client (("The control microprocessor 570 also has programming to keep rule tables for the network processing unit 520 and the classification engine 528 up to date.") column 40 lines 37-38).

Consider claim 5, and as applied to claim 2 above. Garnett et al. discloses a method wherein said control message comprises information to establish a TCP connection between said load balancer and said server (("The application layer parsing and classification engine 528 receives complete request packets requesting the initiation of a new TCP connection from the network processing unit 520.") column 40 lines 3-6).

Consider claim 8. Garnett et al. clearly shows and discloses a computer program product embodied in a machine readable medium for reducing the number of messages to be processed by a control processor in a load balancer comprising the steps of: receiving a request to establish a TCP connection from a client by a network processor in said load balancer (("The network processing unit 520 further includes packet processing circuitry 522 to classify packets at layer 4 and look up flow details of that packet. If a flow exists, the network processing unit 520 modifies and forwards the packet to continue the flow.") column 39 lines 48-52); establishing said TCP connection with said client via handshake messages between said network processor and said client (("For new TCP flows that are to be application layer load balanced, it performs the TCP three-way handshake and forwards the request data packets to an application layer classification engine 528. The network processing unit 520 also handles handover

of a newly established connection to a selected server 43.") column 39 lines 52-58); receiving a request message from said client (("Transport layer load balancing in a TCP/IP network can be performed on receipt of a TCP synchronize/start packet (TCP SYN) from a client to a specific destination.") column 35 lines 9-11); bundling said request message and information from said handshake messages involved in establishing said TCP connection by said network processor (("In HTTP, usually the first data packet contains the request so the first data packet, after the three-way handshake, is parsed to make the load balancing decision. After the load balancing decision has been made, the connection to the load balancer is handed over to the selected server.") column 36 lines 52-56); and transmitting said bundled message to said control processor by said network processor ("When a server responds to a received data packet, the outgoing packet is transmitted from the server 505 to the Load Balancer 501, which forwards the outgoing packet to the outside network. Shown in FIG. 20b is a representation of the data packet paths through a shelf 41 arranged as shown in FIG. 19b. Here, the incoming packet arrives at the switch 73, travels to the load balancer 501 and is then passed to a processing cartridge 43 via the switch 73.") column 33 lines 13-20).

Consider claim 9, and as applied to claim 8 above. Garnett et al. discloses a computer program product comprising the steps of: identifying a server in a server farm to service a client's request message by a control processor (("Management control of the web farm 360 can be provided through a pair of System Management Servers (SMSs) 362.") column 29 lines 23-24); bundling the client's request message and a

control message by the control processor ("Thus incoming data packets arrive at the load balancer and are routed through to a selected server 505") column 32 lines 63-65); and transmitting the bundled message comprising the client's request message and the control message to a network processor ("A control microprocessor 530 performs management tasks for the load balancer 501.") column 40 lines 31-32).

Consider claim 10, and as applied to claim 9 above. Garnett et al. discloses a computer program product wherein said server in said server farm is identified using information extracted from said client's request message (("An information distribution module may be provided removably received in the carrier operable connect to the internal communications network to receive an information message, to perform processing on the message to determine a destination, and to forward the message toward the determined destination via the internal communications network. This arrangement provides a modular computer system with integral load balancing service.") column 1 lines 53-56).

Consider claim 11, and as applied to claim 9 above. Garnett et al. discloses a computer program product wherein said control message comprises information used to enable said network processor to create entries in a forwarding table to ensure packets from said client are transmitted to said server and to ensure packets from said server are transmitted to said client (("The control microprocessor 570 also has programming to keep rule tables for the network processing unit 520 and the classification engine 528 up to date.") column 40 lines 37-38).

Consider claim 12, and as applied to claim 9 above. Garnett et al. discloses a computer program product wherein said control message comprises information to establish a TCP connection between said load balancer and said server ((“The application layer parsing and classification engine 528 receives complete request packets requesting the initiation of a new TCP connection from the network processing unit 520.”) column 40 lines 3-6).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in Graham v. John Deere Co., 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were

made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 6-7 and 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Garnett et al. (US 7032037 B2) in view of Berg (US 20020116475 A1).

Consider claim 6, and as applied to claim 2 above. Garnett et al. clearly shows and discloses a method for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed "The method as recited in claim 2 further comprising the steps of: receiving a request to terminate said TCP connection from said server by said network processor; facilitating said termination of said connection between said server and said client; ..." (("In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach bundling and transmitting information of closed connections. Berg discloses a protocol stack comprising temporary memory tables, to include connection information and a method of transmitting session

management packet bundles from a redirector device to a network interface controller. This reads on the claimed "... bundling information regarding a series of closed connections by said network processor; and transmitting said bundled message regarding said series of closed connections to said control processor by said network processor." (("If the packet is not a client request (e.g. TCP Flag set to ACK), the iNIC (in response to instructions of its balance thread) sends the packet and a reference to the connection endpoint (stored in the temporary table's matching record) to the protocol stack thread (which is executed by the iNIC's protocol stack processor).") paragraph 0159 ("Each of the n servers and the redirector device includes intelligent network interface controller ("iNIC") circuitry, as shown in FIG. 2a. Within the server farm, each of the n servers and the redirector device (with its respective iNIC) has a respective IP address that is advertised to clients through the IP network. The redirector device and the servers communicate with one another through the iNICs, in order to operate together in a cooperative manner as a distributed system.") paragraph 0064).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate protocol stacks as taught by Berg with TCP termination requests as taught by Garnett et al. for the purpose of session management using the well known TCP stack.

Consider claim 7, and as applied to claim 6 above. Garnett et al. clearly shows and discloses a method for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed "... information from said bundled message regarding said series of closed connections

by said control processor." (("In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach a method of extracting bundled information of closed connections. Berg discloses a method wherein packet information is extracted and exchanged between a client and a server. This reads on the claimed "The method as recited in claim 6 further comprising the step of: extracting information from said bundled message regarding said series of closed connections by said control processor." (("Accordingly, in such a situation, server 2's iNIC (in response to instructions of its ipOS): (a) in response to such information received from server 1's iNIC establishes a connection endpoint in the memory of server 2's iNIC for the particular client-server socket-based application connection; (b) if appropriate for the packet, processes and sends information from the packet to server 2's application layer; and (c) if appropriate for the packet, processes and sends response packets to the client through the IP network in response to information from server 2's application layer. The protocol stack processor of server 2's iNIC (in response to instructions of its ipOS) adds suitable header information to the response packet and sends it to the client through the IP network-connected port (IP 123.123.123.3) of server 2's iNIC.") paragraph 0102).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate packet processing as taught by Berg with TCP termination requests as taught by Garnett et al. for the purpose of releasing closed TCP connections faster, and providing more resources for new connections.

Consider claim 13, and as applied to claim 9 above. Garnett et al. clearly shows and discloses a computer program for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed "The method as recited in claim 2 further comprising the steps of: receiving a request to terminate said TCP connection from said server by said network processor; facilitating said termination of said connection between said server and said client; ..." (("In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach bundling and transmitting information of closed connections. Berg discloses a protocol stack comprising temporary memory tables, to include connection information and a method of transmitting session management packet bundles from a redirector device to a network interface controller. This reads on the claimed "... bundling information regarding a series of closed connections by said network processor; and transmitting said bundled message

regarding said series of closed connections to said control processor by said network processor." (("If the packet is not a client request (e.g. TCP Flag set to ACK), the iNIC (in response to instructions of its balance thread) sends the packet and a reference to the connection endpoint (stored in the temporary table's matching record) to the protocol stack thread (which is executed by the iNIC's protocol stack processor).") paragraph 0159 ("Each of the n servers and the redirector device includes intelligent network interface controller ("iNIC") circuitry, as shown in FIG. 2a. Within the server farm, each of the n servers and the redirector device (with its respective iNIC) has a respective IP address that is advertised to clients through the IP network. The redirector device and the servers communicate with one another through the iNICs, in order to operate together in a cooperative manner as a distributed system.") paragraph 0064).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate protocol stacks as taught by Berg with TCP termination requests as taught by Garnett et al. for the purpose of session management using the well known TCP stack.

Consider claim 14, and as applied to claim 13 above. Garnett et al. clearly shows and discloses a computer program for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed "... information from said bundled message regarding said series of closed connections by said control processor." (("In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish

packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach a computer program of extracting bundled information of closed connections. Berg discloses a computer program wherein packet information is extracted and exchanged between a client and a server. This reads on the claimed "The method as recited in claim 6 further comprising the step of: extracting information from said bundled message regarding said series of closed connections by said control processor." (("Accordingly, in such a situation, server 2's iNIC (in response to instructions of its ipOS): (a) in response to such information received from server 1's iNIC establishes a connection endpoint in the memory of server 2's iNIC for the particular client-server socket-based application connection; (b) if appropriate for the packet, processes and sends information from the packet to server 2's application layer; and (c) if appropriate for the packet, processes and sends response packets to the client through the IP network in response to information from server 2's application layer. The protocol stack processor of server 2's iNIC (in response to instructions of its ipOS) adds suitable header information to the response packet and sends it to the client through the IP network-connected port (IP 123.123.123.3) of server 2's iNIC.") paragraph 0102).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate packet processing as taught by Berg with TCP termination requests as

taught by Garnett et al. for the purpose of releasing closed TCP connections faster, and providing more resources for new connections.

Claims 15-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Garnett et al. (US 7032037 B2) in view of Basso et al. (US 6973503 B2).

Consider claim 15. Garnett et al. clearly shows and discloses a load balancer ((“The present invention relates to computer systems, in particular to load balancing systems for multi-processor systems, for example multi-processor server systems.”) column 1 lines 10-12) comprising a network processor capable of responding to a computer program with circuitry for receiving a request to establish a TCP connection from a client by a network processor in said load balancer ((“The network processing unit 520 further includes packet processing circuitry 522 to classify packets at layer 4 and look up flow details of that packet. If a flow exists, the network processing unit 520 modifies and forwards the packet to continue the flow.”) column 39 lines 48-52); establishing said TCP connection with said client via handshake messages between said network processor and said client ((“For new TCP flows that are to be application layer load balanced, it performs the TCP three-way handshake and forwards the request data packets to an application layer classification engine 528. The network processing unit 520 also handles handover of a newly established connection to a selected server 43.”) column 39 lines 52-58); receiving a request message from said client ((“Transport layer load balancing in a TCP/IP network can be performed on receipt of a TCP synchronize/start packet (TCP SYN) from a client to a specific destination.”) column 35 lines 9-11); bundling said request message and information

from said handshake messages involved in establishing said TCP connection by said network processor (("In HTTP, usually the first data packet contains the request so the first data packet, after the three-way handshake, is parsed to make the load balancing decision. After the load balancing decision has been made, the connection to the load balancer is handed over to the selected server.") column 36 lines 52-56); and transmitting said bundled message to said control processor by said network processor ("When a server responds to a received data packet, the outgoing packet is transmitted from the server 505 to the Load Balancer 501, which forwards the outgoing packet to the outside network. Shown in FIG. 20b is a representation of the data packet paths through a shelf 41 arranged as shown in FIG. 19b. Here, the incoming packet arrives at the switch 73, travels to the load balancer 501 and is then passed to a processing cartridge 43 via the switch 73.") column 33 lines 13-20). This reads on the claimed "A load balancer, comprising: a network processor, wherein said network processor is configured to process fast path packets; a control processor coupled to said network processor, wherein said control processor is configured to process slow path packets; and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor; wherein said network processor, responsive to said computer program, comprises: circuitry operable for receiving a request to establish a TCP connection from a client; circuitry operable for establishing said TCP connection with said client via handshake messages between said network processor and said client; circuitry operable for receiving a request

message from said client; circuitry operable for bundling said request message and information from said handshake messages involved in establishing said TCP connection; and circuitry operable for transmitting said bundled message to said control processor.". However, Garnett et al. fails to teach a system comprising fast path and slow path packet processing. Basso et al. discloses a system wherein a network processor processes fast packets and a control processor processes slow packets ((Network processor 411 may be configured to process packets that are commonly referred to as "fast path packets.") column 5 lines 58-60 ("... control processor 410 may be configured to process packets that are commonly referred to as "slow path packets" which require more complicated operations than fast path packets. Slow path packets may refer to packets that are redirected from network processor 411 to control processor 410 to be processed by control processor ...") column 5 line 67 and column 6 lines 1-5), and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor (("Implementations of the invention include implementations as a computer system programmed to execute the method or methods described herein, and as a computer program product. According to the computer system implementations, sets of instructions for executing the method or methods are resident in the random access memory 414 of one or more computer systems configured generally as described ...") column 6 lines 60-67). This reads on the claimed "... a network processor, wherein said network processor is configured to process fast path packets; a control processor

coupled to said network processor, wherein said control processor is configured to process slow path packets; and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor; ..."

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate fast path and slow path packet processing as taught by Basso et al. with bundling messages in TCP connections as taught by Garnett et al. for the purpose of efficiently processing packets in a well known TCP protocol connection.

Consider claim 16, and as applied to claim 15 above. Garnett et al., as modified by Basso et al. clearly shows and discloses a method comprising the steps of: identifying a server in a server farm to service a client's request message by a control processor (("Management control of the web farm 360 can be provided through a pair of System Management Servers (SMSs) 362.") column 29 lines 23-24); bundling the client's request message and a control message by the control processor ("Thus incoming data packets arrive at the load balancer and are routed through to a selected server 505") column 32 lines 63-65); and transmitting the bundled message comprising the client's request message and the control message to a network processor ("A control microprocessor 530 performs management tasks for the load balancer 501.") column 40 lines 31-32).

Consider claim 17, and as applied to claim 16 above. Garnett et al., as modified by Basso et al. clearly shows and discloses a method wherein said control message comprises information used to enable said network processor to create entries in a forwarding table to ensure packets from said client are transmitted to said server and to ensure packets from said server are transmitted to said client ((“The control microprocessor 570 also has programming to keep rule tables for the network processing unit 520 and the classification engine 528 up to date.”) column 40 lines 37-38).

Claims 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Garnett et al. (US 7032037 B2) as modified by Basso et al. (US 6973503 B2) and in further view of Berg (US 20020116475 A1).

Consider claim 18. Garnett et al. clearly shows and discloses a method for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed “... method ... comprising the steps of: receiving a request to terminate said TCP connection from said server by said network processor; facilitating said termination of said connection between said server and said client; ...” ((“In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer

accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach a system comprising fast path and slow path packet processing. Basso et al. discloses a system wherein a network processor processes fast packets and a control processor processes slow packets ((Network processor 411 may be configured to process packets that are commonly referred to as "fast path packets.") column 5 lines 58-60 ("... control processor 410 may be configured to process packets that are commonly referred to as "slow path packets" which require more complicated operations than fast path packets. Slow path packets may refer to packets that are redirected from network processor 411 to control processor 410 to be processed by control processor ...") column 5 line 67 and column 6 lines 1-5), and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor (("Implementations of the invention include implementations as a computer system programmed to execute the method or methods described herein, and as a computer program product. According to the computer system implementations, sets of instructions for executing the method or methods are resident in the random access memory 414 of one or more computer systems configured generally as described ...") column 6 lines 60-67). This reads on the claimed "... a network processor, wherein said network processor is configured to process fast path packets; a control processor coupled to said network processor, wherein said control processor is configured to process slow path packets; and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer

program for reducing the number of messages to be processed by said control processor; ..." Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate fast path and slow path packet processing as taught by Basso et al. with bundling messages in TCP connections as taught by Garnett et al. for the purpose of efficiently processing packets in a well known TCP protocol connection. However, Garnett et al., as modified by Basso et al., fails to teach bundling and transmitting information of closed connections. Berg discloses a protocol stack comprising temporary memory tables, to include connection information and a method of transmitting session management packet bundles from a redirector device to a network interface controller. This reads on the claimed "... bundling information regarding a series of closed connections by said network processor; and transmitting said bundled message regarding said series of closed connections to said control processor by said network processor." (("If the packet is not a client request (e.g. TCP Flag set to ACK), the iNIC (in response to instructions of its balance thread) sends the packet and a reference to the connection endpoint (stored in the temporary table's matching record) to the protocol stack thread (which is executed by the iNIC's protocol stack processor).") paragraph 0159 ("Each of the n servers and the redirector device includes intelligent network interface controller ("iNIC") circuitry, as shown in FIG. 2a. Within the server farm, each of the n servers and the redirector device (with its respective iNIC) has a respective IP address that is advertised to clients through the IP network. The redirector device and the servers communicate with one another through the iNICs, in order to operate together in a cooperative manner as a distributed system.") paragraph 0064).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate protocol stacks as taught by Berg with TCP termination requests as taught by Garnett et al., as modified by Basso et al., for the purpose of session management using the well known TCP stack.

Consider claim 19, and as applied to claim 18 above. Garnett et al. clearly shows and discloses a method for receiving a request to terminate a TCP session and facilitating the termination request between server and client. This reads on the claimed "... information from said bundled message regarding said series of closed connections by said control processor." (("In the present example, when a flow is finally terminated the load balancer deletes the entry from the primary look-up table which corresponded to that connection. Thus the load balancer can watch for a TCP finish packet (TCP FIN) to be received from the client and a corresponding ACK to be sent from the server. In the case of a triangular load balancing arrangement, a module running on the server may perform the watch for the FIN and ACK packets and notify the load balancer accordingly.") column 36 lines 14-22). However, Garnett et al. fails to teach a system comprising fast path and slow path packet processing. Basso et al. discloses a system wherein a network processor processes fast packets and a control processor processes slow packets ((Network processor 411 may be configured to process packets that are commonly referred to as "fast path packets.") column 5 lines 58-60 ("... control processor 410 may be configured to process packets that are commonly referred to as "slow path packets" which require more complicated operations than fast path packets. Slow path packets may refer to packets that are redirected from network processor 411

to control processor 410 to be processed by control processor …") column 5 line 67 and column 6 lines 1-5), and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor (("Implementations of the invention include implementations as a computer system programmed to execute the method or methods described herein, and as a computer program product. According to the computer system implementations, sets of instructions for executing the method or methods are resident in the random access memory 414 of one or more computer systems configured generally as described …") column 6 lines 60-67). This reads on the claimed "The system as recited in claim 18 comprising a network processor, wherein said network processor is configured to process fast path packets; a control processor coupled to said network processor, wherein said control processor is configured to process slow path packets; and a memory unit coupled to said control processor and said network processor, wherein said memory unit is operable for storing a computer program for reducing the number of messages to be processed by said control processor; ..." Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate fast path and slow path packet processing as taught by Basso et al. with bundling messages.in TCP connections as taught by Garnett et al. for the purpose of efficiently processing packets in a well known TCP protocol connection. However, Garnett et al., as modified by Basso et al., fails to teach a method of extracting bundled information of closed connections. Berg discloses a method wherein packet information is extracted and exchanged

between a client and a server. This reads on the claimed "The method as recited in claim 6 further comprising the step of: extracting information from said bundled message regarding said series of closed connections by said control processor." (("Accordingly, in such a situation, server 2's iNIC (in response to instructions of its ipOS): (a) in response to such information received from server 1's iNIC establishes a connection endpoint in the memory of server 2's iNIC for the particular client-server socket-based application connection; (b) if appropriate for the packet, processes and sends information from the packet to server 2's application layer; and (c) if appropriate for the packet, processes and sends response packets to the client through the IP network in response to information from server 2's application layer. The protocol stack processor of server 2's iNIC (in response to instructions of its ipOS) adds suitable header information to the response packet and sends it to the client through the IP network-connected port (IP 123.123.123.3) of server 2's iNIC.") paragraph 0102).

Therefore, it would have been obvious for a person of ordinary skill in the art to incorporate packet processing as taught by Berg with TCP termination requests as taught by Garnett et al., as modified by Basso et al., for the purpose of releasing closed TCP connections faster, and providing more resources for new connections.

Conclusion

Any response to this Office Action should be faxed to (571) 273-8300 or mailed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand-delivered responses should be brought to

Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Mark Fearer whose telephone number is (571) 270-1770. The Examiner can normally be reached on Monday-Thursday from 7:30am to 5:00pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Rafael Pérez-Gutiérrez can be reached on (571) 272-7915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 571-272-4100.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-2600.

Mark Fearer
M.D.F./mdf

March 12, 2007